A Low-Cost, Compact, Block-Mode Computer Terminal

The design emphasizes ergonomics and very high reliability as well as low cost and compactness.

by Jean-Louis Chapuis and Michèle Prieur

IN 1981, HEWLETT-PACKARD introduced the HP 2622A Terminal, which was the basic product of the HP 262X family. At that time, the low-cost terminals war had already started, and many of HP's competitors were offering new character- and block-mode terminals at very aggressive prices. A market study showed that more and more customers were becoming sensitive to the cost of their video display units, because they represent a significant part of the total cost of a computer system.

In response, the HP 2392A Terminal project was started at HP's Grenoble Personal Computer Division. The project had very ambitious objectives:

- Cut the manufacturing cost of the previous terminal in half
- Develop a new package design emphasizing ergonomics and compactness
- Achieve a reliability equivalent to less than a single failure every ten years.

It was realized early in the project that only the use of specialized VLSI components would drastically reduce the component costs and allow the price and reliability objectives to be met. Two VLSI chips were designed. One is a CRT controller manufactured at HP's Cupertino Integrated Circuits Operation (see article, page 9). The other is an ECL gate array developed with an outside vendor. As a result, the HP 2392A (Fig. 1) has one fourth as many chips on its logic board as its predecessor (see Fig. 2), yet offers greatly improved performance and features.

Even the power supply of this new terminal was an in-house design. This was because no commercially available power unit could match the stringent requirements for reliability, compactness, and price.

Ergonomics requirements dictated a tilt and swivel capability. The traditional method of providing the tilt feature has been an articulated pedestal that tilts the entire unit. This method adds substantial height to the terminal and increases its shipping bulk. The solution was to hinge the CRT tube and tilt the tube itself with respect to the terminal housing. The external dimension constraints imposed by HP's corporate design standardization program (width of 325 mm) and the internal space required by the tilting tube left little room for electronics. Various attempts at the layout and arrangement of the power supply and sweep printed circuit boards were made to facilitate heat dissipation and avoid a cooling fan, which is highly undesirable in the office environment.

A very close relationship with HP's U.S. divisions was required to ensure a good integration of the terminal with HP computers, software, and printers. On the other hand, close collaboration was necessary with European countries to localize the product.

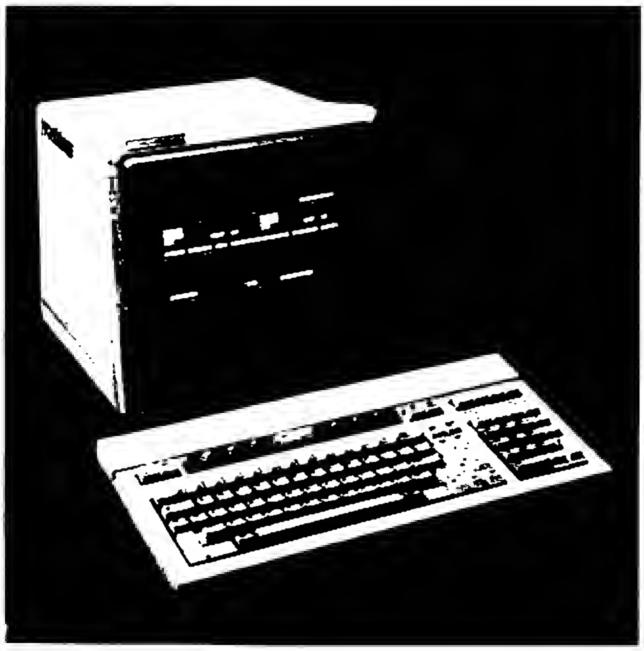

Fig. 1. *HP 2392A Terminal.*

Fig. 2. *The HP 2392A's logic board (right) has only one fourth as many chips as earlier HP designs (left).*

Feature Set and Firmware Design

Compatibility with existing HP block-mode terminals was of paramount importance for the HP 2392A. Therefore, the HP 2392A firmware is a superset of the firmware of its predecessor, the HP 2622A. Additional features include smooth scrolling, a parallel or serial printer port, dynamic allocation of display memory, secret video enhancement, and datacom speeds up to 19,200 baud

The HP 2392A can display softkey labels and error messages in ten languages and can handle 17 national keyboards. It supports the Roman8 character set standard, which includes both USASCII characters and international characters. The 256 Roman8 characters are represented by 8-bit codes.

The Parity/Data Bits field in the datacom configuration menu allows the user to specify whether character codes are sent as 7-bit codes or as 8-bit codes. If the 8-bit mode is selected, international characters are accessed directly rather than being taken from an alternate character set, which is accessed indirectly through additional control code. If the 7-bit mode is selected, the terminal uses the ISO replacement method, which is compatible with the HP 2622A's 7-bit mode.

Some functions normally performed by hardware are partially done by the HP 2392A firmware. For instance, keyboard encoding and debouncing uses a minimum of hardware (see box, page 7).

The smooth scrolling of the display is also partially handled in an NMI (nonmaskable interrupt) routine, where a smooth scroll counter is maintained. The video structure is updated according to this counter.

Video Structure

The display memory uses dynamic allocation, which means that characters to be displayed are stored in memory blocks taken from a linked list of free blocks. The video structure is composed of linked row headers, which point to data blocks that contain the row data.

One normal row is composed of a 16-byte header and from one to four 80-byte data blocks.

When a line is empty, only a header of 16 bytes is used. Whenever one character is entered in this row, an 80-byte block is fetched from the pool of free blocks and is allocated to this header. Thus, 16 + 80 = 96 bytes are required to display a complete row of normal characters.

If some video attributes (e.g., inverse video, blinking, etc.) or some software attributes (unprotected fields in format-mode applications) are required, then two more 80-byte blocks are fetched from the pool and linked to the row header using the attribute addresses in the header, as shown in Fig. 3. Thus, to display a row with hardware and software attributes, 16 + 3 × 80 = 256 bytes are used.

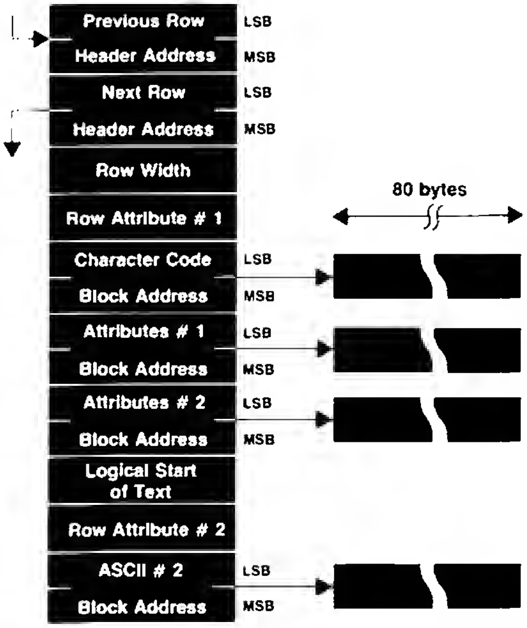

Fig. 3. *In the HP 2392A's display memory, linked row headers point to data blocks that contain the characters for each row.*

The last block address shown in Fig. 3, the ASCII #2 block address, is used when a secret video enhancement is required in the row. The logical start of text is used in the line modify and modify all modes.

The HP 2392A can display up to two pages of text even in the worst-case situation (full 80-character rows with attributes). In a typical case, the terminal can display up to four pages. If no attributes are required, it can display four pages. Additional RAM can be ordered as an option, allowing the terminal to display up to eight pages.

Because of dynamic allocation, a situation may arise where no more 16-byte headers are available from the 16-byte block pool. In such a situation, 80-byte blocks are fetched from the free 80-byte block pool and are used by the terminal firmware to create new row headers.

Block Mode

The block-mode tear-apart firmware handles transfer requests. It is divided into three main parts: request handling, handshake handling, and actual transfers. Each of these parts is independent of the others.

When a transfer is requested, a routine is called that decides the type of transfer requested. Then another routine decides the type of handshake to be used and whether the transfer is to be performed immediately or put in a waiting state. Fig. 4 shows the sequence of operations.

Configuration

The various terminal characteristics can be configured easily by displaying and modifying the configuration menus on the screen. The user can change the contents of these menus and then alter the terminal's configuration characteristics by saving them in nonvolatile CMOS memory, which is mapped in a part of the terminal memory called CMOS-Image. For security reasons, this part of the terminal RAM is duplicated in two different areas of the nonvolatile RAM.

These areas, called CMOS1 and CMOS2, are only modified by the SAVE key or by certain softkeys, whose states are also saved in this memory. The softkeys whose states are saved in the CMOS memory are: remote mode, auto linefeed, block mode, modify all, to external, and to display.

At power-on, CMOS1 is copied into CMOS-Image. It provides information on how the terminal was last configured. CMOS-Image contains the current state of the terminal. This part of RAM can be modified by escape sequence or by the SAVE key.

A CMOS scratch RAM is used to display the configuration menu. A scratch area is necessary since the user may change the menu and decide not to save it.

When it is turned on for the first time, the terminal is configured with default values.

Parser

Every character received is interpreted as a command or part of a command and triggers the appropriate action. In terms of language theory, these commands are phrases obeying the rules of a regular grammar, and therefore may be recognized by a finite state machine.

The HP 2392A's state machine consists of a control unit (CU) and a language descriptor (LD). The latter is a collection of data describing the commands accepted by the terminal. The former is a very short program, able to execute the elementary orders issued by the LD. Both are designed to take advantage of the separation of commands into families. This makes the state machine modular and easy to change. Only, the LD has to be modified or rewritten if new commands or applications are implemented.

In the LD, the elementary building block is the transition. When it receives a character, the state machine executes a move from an initial state to a final state, and triggers an action. This is a transition. Several transitions are possible for a given initial state, depending on the input character.

A state is the collection of all the transitions issued from it. These transitions must be contiguously listed in memory, but their order is irrelevant. If the first transition does not match the input character, the following transition is tried, and so on until a match is found. Thus, the last transition must collect the unexpected input characters.

A family is a collection of related states. It is represented by a list containing the addresses of these states.

An application is a collection of families. It is represented by a list containing the addresses of these families.

The CU is designed with reentrance capability to allow several applications to run simultaneously. Every application has a dedicated workspace for its variables.

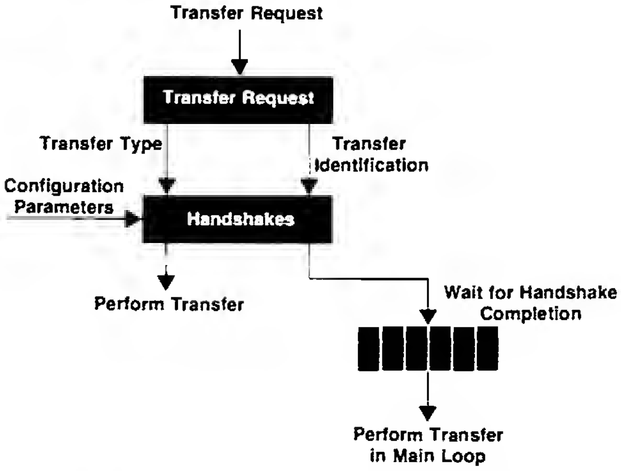

Fig. 4. *Block mode transfer handling in the HP 2392A.*

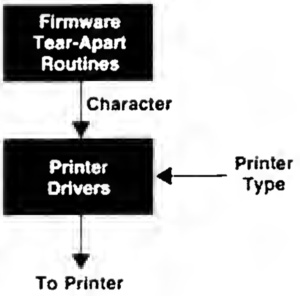

Fig. 5. *For output to an external printer, one of four printer drivers is chosen, depending on the printer type.*

External Port

The routines that handle data transfers to the external port are in fact the routines of the tear-apart firmware with small modifications. These routines get characters from the video RAM and send them to a driver, which transforms them if necessary so they can be understood by the printer connected to datacom port 2 (Fig. 5).

There are four printer drivers, one of which is chosen according to the printer type. The terminal recognizes two types of printers: HP printers (HP 267X) and others. Both kinds of printers have USASCII character sets, but international characters and alternate character sets (e.g., line drawing set) are handled differently. Both types of printers also have 7-bit and 8-bit modes.

The Other_7bits driver uses ISO replacement for the international characters and accesses the line drawing set as an alternate character set. The Other_8bits driver implements 8-bit datacom. With the 267X_7bits driver, the Roman extension and line drawing sets are considered as alternate sets. They are accessed via escape sequences and control codes. With the 267X_8bits driver, the Roman extension and line drawing sets are also considered as alternate character sets, but they are accessed via escape sequences and bit 8 is set to 1.

Test Programs

The HP 2392A has three types of built-in test programs: Power-on tests, self-tests, and manufacturing test. Power-on tests are executed at power-on. These test the ROMs and the RAMs (including the CMOS RAM). If no errors are detected, control is passed to the main program. Otherwise, the terminal gives a number of beeps corresponding to the test that failed and the program stops.

There are three self-tests: terminal test, port 1 test, and port 2 test. The terminal test tests the ROMs and RAMs and displays the test pattern on the screen. The port 1 test and the port 2 test differ only by the address of the hardware handshake lines. During these tests, serial transmission and the hardware handshake lines are tested.

The manufacturing test is performed at power-on if a special module is plugged in. This test has three different modes. The first mode is a complete test of the hardware (ROMs, RAMs, interrupt lines, port 1, and port 2). Errors are displayed on LEDs located on the special module. This program can run with a minimum hardware configuration.

A Reliable, Low-Cost Keyboard Interface

Low-cost communication between a detachable keyboard and a terminal processor board can be achieved using a simple hardware software procedure. Traditional keyboard interface circuits rely generally on data transfer between the keyboard itself and the processor board. Either serial or parallel data transfer may be used. Serial transfer requires four lines: two for power, one for data, and one sync line. Specialized chips are often used, along with other components, to transmit the serial data stream. On the other hand, parallel data transfer requires at least eight data lines, one strobe, and two power lines.

To reduce component count and overall interface cost, the HP 2392A uses a different approach originally developed for the HP 2621B Display Terminal and later used in the HP 150 Personal Computer. The main advantage of this method is that it does not require the transmission of key number information, either coded or not. The hardware in the keyboard consists of column drivers, row scanners, and a 7-bit counter to address up to 128 key locations (see Fig. 1). On the processor board, one byte in memory is dedicated as the image of the counter in the keyboard. A five-wire, low-cost telephone cable is used for the physical connection, allowing for two power lines and three control lines: increment, reset, and key state.

The principle of operation is straightforward. It is based on the identity of the contents of the counter in the keyboard and its memory image on the processor side. The scanning routine in the processor board clears the image byte in memory and issues a reset to the interface, effectively clearing the keyboard counter. The key state line is sampled to check the status of key location 0. The scanning routine then increments the image byte in memory and issues an increment to the interface. Key location 1 is addressed, and its status is read on the key state line. This process is repeated until the last key location on the keyboard has been checked, the reset line is activated again, and the image byte is cleared. Any time a key is found in the depressed state, the scanning routine looks in the image byte to find the key location.

Khambao Panyasak
Development Engineer
Grenoble Personal Computer Division

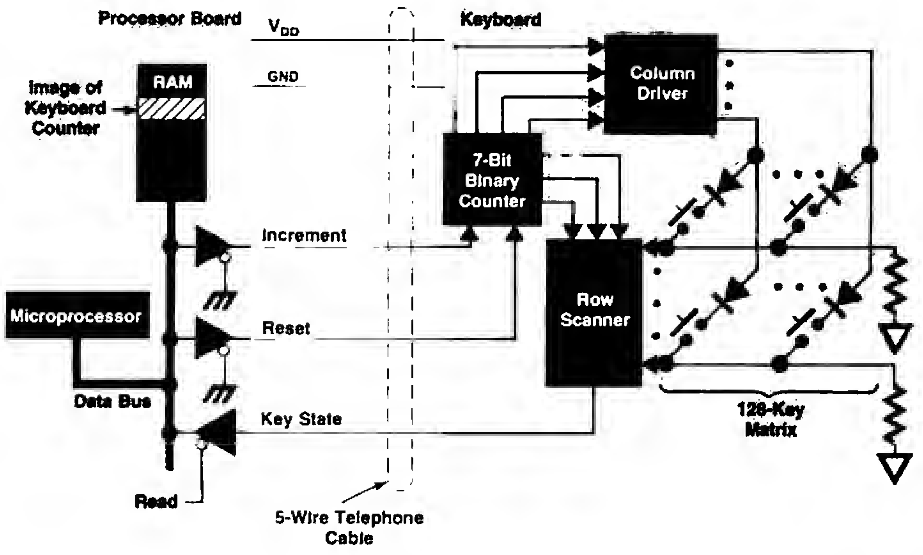

Fig. 1. *Diagram of the low-cost connection scheme between the keyboard and the processor of the HP 2392A Terminal*

This test is performed continuously in a loop. The second mode is identical to the first, but the test is performed only once. In the third mode, the terminal only fills the screen witb @ to allow the operator to adjust the screen during production.

Acknowledgments

Several persons worked on the HP 2392A firmware, Nicolas Tripon was in charge of the parser and the datacom code. Nadine Odet worked mainly on the configuration and the softkey parts. Michel Ghidossi wrote all of the test modules and Freddie Barbut was in charge of the keyboard code. We would like to thank all of the people who helped us test this software for HP 2622A compatibility on application programs, and especially Jacques Duharest.

Mechanical Design of a Low-Cost Terminal

by Michel Cauzid

THE HP 2392A TERMINAL is the first HP display product that conforms to a new HP industrial design program for computer products. This program has the objective of giving different computer products consistent shapes, dimensions, and colors. Consistent dimensions allow products to be used side by side or on top of one another. A new range of colors, paler than those used previously, has been introduced, in tune with a trend towards lighter colors for office equipment and furniture.

The program also emphasizes consistent implementation of good human factors features for display products..This includes such things as integral display tilt and swivel mechanisms, detached low-profile keyboard, and terminal controls on the front panel.

External Design

Two new colors are used for the HP 2392A Terminal. French gray is used for the bezel, and parchment white is used for all the other parts. A new smoother texture is employed for all external areas. Attention has been paid to the choice of these colors and texture to minimize the reflection of light on the monitor housing.

The main purpose of screen tilt and swivel is to provide an optimum viewing angle for operators of different size, allowing them to achieve the most comfortable posture. The operator can tilt the terminal to reduce glare from ambient lighting. The contribution of the HP 2392A in this area is the integration of the tilt movement in the enclosure. Internal tilt allows simple vertical adjustment of the CRT without having to move the entire enclosure. The nominal tilt range is 0 degrees vertical to 20 degrees back. The swivel in the base allows 360 degrees of rotation.

All user controls are located on the front, below the CRT. The power button is located on the lower left portion of the display bezel. Brightness control is located under the lower right portion of the display bezel.

The HP 2392A is designed for minimum size and maximum configurational flexibility. It is a member of the 325-mm stack family of products defined by the new industrial design programs. In this family, all peripherals, such as printers, are external to the terminal enclosure. The overall width of the terminal is 325 mm. Since the 12-in CRT's width is 275 mm, only 25 mm is available on each side for the tilt mechanism.

Internal Design

An internal metallic chassis holds the logic, power supply, and sweep boards, as well as the screen tilt mechanism. This sheet-metal part forms a rigid frame for the terminal. All the architecture is based around the sheet-metal frame. This permitted more flexibility during the tooling phase.

The CRT is fastened to the moving bezel with two metallic brackets. This assembly tilts around two holes in the metallic chassis. It is balanced by a spring and braked by friction. Fig. 1 is a drawing of the terminal, showing the tilt mechanism.

Three other plastic parts make an aesthetic enclosure: the cover, the front bezel, and the plastic chassis. Under the plastic chassis, a pedestal with rubber feet provides the swivel function.

The plastic molding compound used for the packaging is an ABS from Borg Warner Company (Cycolac KJUV). For friction compatibility between the pedestal and chassis, the pedestal is molded with Makrolon, a polycarbonate

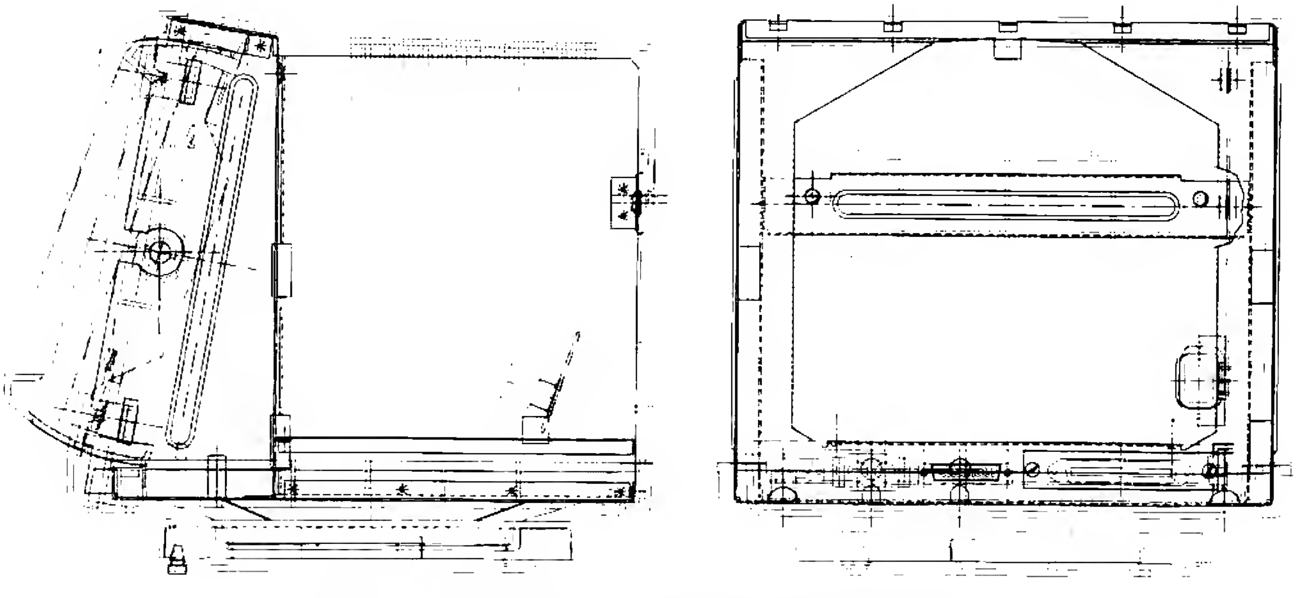

Fig. 1. *An HP Draft drawing of the HP 2392A CRT tilt mechanism.*

from Bayer.

Internal connections are minimized. The only connections using wires are between the CRT and the boards. All the other connections are directly on the boards.

Five Design Phases

The mechanical design team was relatively small. During the five design phases, this team had to design the parts, draw the schematics, choose the subcontractors, assist in the tooling setup, and perform the tests. These tasks were made easier by the use of HP Draft (see below).

During the conceptual phase, the team was responsible for conceiving and defining the physical parts used in the product. The objectives were to reduce stock levels, minimize inventory problems, reduce the assembly time, and use standard circuit sizes. For the materials choice, we looked for competitively priced materials with high-quality characteristics. We designed all parts as tooled parts and we subcontracted the production of all parts.

The next phase was the documentation phase. The basic three-view mechanical drawing is the primary document that directs people in producing each part. Each member of the team designed parts and created drawings using HP Draft. The simulation capabilities of HP Draft guaranteed that the parts would fit together properly.

The next phase was the modeling phase. Usually a designer first makes mechanical drawings of a new part, and then to verify the design, sends the drawings to a model shop to have one or two parts made. For the HP 2392A, we were able to go directly from the first drawings to molds. We subcontracted to an external model shop the manufacture of soft molds. With these tools we obtained 100 parts using a new process based on low-pressure injection. Once painted, these parts look very similar to traditional injection-molded parts.

The fourth mechanical design phase was the tooling phase. The HP 2392A has five major tooled pastic parts, five major tooled sheet-metal parts, and ten miscellaneous tooled metal or plastic parts. The production of these parts is subcontracted. The ideal subcontractor from our point of view is the one who can combine efficient production with enough flexibility to be on our design team. This means reacting quickly in case of modifications to avoid wasting valuable time. To meet our schedule, we finalized the plastic parts first, since modifying sheet metal is easier than modifying molds. All the molds are designed so that the parts can be injected in polycarbonate or in ABS.

In the test phase, our major problem was the behavior of the moving CRT during the shock and vibration tests, and the efficiency of the package with this tilt mechanism. In addition to the standard environmental tests we tested the tilt mechanism extensively to verify the choice of the friction material.

HP Draft

HP Draft is a software program that runs on HP 9000 Model 520 and HP 9845B Computers. It facilitates the production of two-dimensional mechanical drawings by working interactively with the designer, who has great flexibility in specifying positions, points, intersections, objects, and so on. Fig. 1 of this article is an example of an HP Draft drawing.

The HP 2392A project was started using an HP 9845B Desktop Computer and an HP 7580A Plotter. At the end, the equipment included three HP 9845Bs, an HP 7580A, and an HP 7585A Plotter.

It is very difficult to estimate the amount of time saved by using HP Draft, but it is undoubtedly true that both the quality and the speed of obtaining final drawings were considerably increased. HP Draft permitted a simulation of the assembly of all parts, and hence saved time and enhanced quality.

HP Draft was especially useful in dealing with our subcontractors. We quickly supplied them with accurate and reliable drawings when design changes occurred.

Acknowledgments

The mechanical design team included Claude Carpentier, industrial designer Jacques Firdmann, Jean-Claude Serres, and Hugues de Charentenay.

VLSI Design in the HP 2392A Terminal

by Jean-Jacques Simon

WHEN HP'S GRENOBLE Personal Computer Division received the charter to design the first of a new generation of display terminals, the objective was to design a terminal that was more powerful, more reliable, and smaller than the previous generation of equivalent products, and that cost half as much.

An analysis showed that the only way to achieve this was to integrate the CRT display controller function. Commercially available controllers did not provide all the functions required by the terminal, and the use of standard MSI chips would consume a large portion of the available board space. A comparison with existing terminals, like the HP 2622A, showed that about 30 MSI or SSI packages could be replaced by a single custom IC. By doing so, the cost of the CRT control function could be reduced by more than 80%.

One of the main objectives was low price. We had to select an integrated circuit process that would yield low-cost parts and that was readily available, well supported, and relatively easy to design with. Among the HP processes available at that time, NMOS-C, manufactured at two locations in the U.S.A., was chosen.

CRT Controller Functions

The HP 2392A's CRT controller (CRTC) is in charge of all the display support functions. Its internal RAM, sequencer PLAs,* and character ROM allow it to fetch ASCII codes from the processor's RAM, store them locally for the duration of a character row on the screen, and convert them to the dot pattern representing the character shape. All these functions are performed with minimum help from the processor, since the CRTC works in a DMA (direct memory access) mode. The CRTC also provides the synchronization signals required to operate the CRT.

The logic board of the HP 2392A can be described with the block diagram in Fig. 1. It is essentially an 8088 microprocessor with ROM and RAM space, a UART,* clock generation logic, and the CRTC. The 8088 and the CRTC share the same RAM space. When characters are entered though the keyboard or received from the datacom port, the microprocessor stores them in the display RAM area. On every vertical retrace time, an NMI (nonmaskable interrupt) is issued to the processor, which executes a routine that writes frame information into the CRTC's register stack.

One of the CRTC registers is the top header address (THA) register. In this register, the microprocessor writes the address of the first block of pointers in the linked list of pointers for the characters to be displayed in the current frame (see Fig. 2). The master sequencer in the CRTC is programmed to use the THA as the first address for a DMA sequence during which it fetches the contents of the top header pointer block. The pointer block, or row header, as described in Fig. 3, contains enough information to find the data and attributes blocks for the current row of characters, as well as the address of the next pointer block, which is used during the following DMA cycle. The row width register in the pointer block indicates the number of displayed characters in the row to be fetched. The DMA cycle is adjusted for that number of characters, reducing bus use by the CRTC. Valid row width values are between 0 and 79 for numbers of characters between 1 and 80. If the row width code is set to 99, the current row on the display is empty, and there is no DMA at all (the row buffer in the chip is filled with space characters).

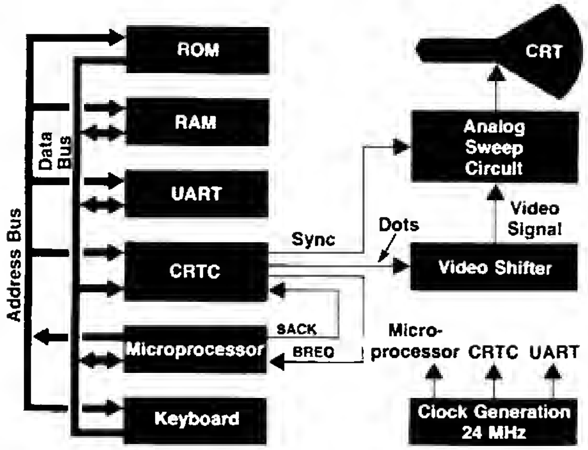

Fig. 1. *Logic block diagram of the HP 2392A Terminal.*

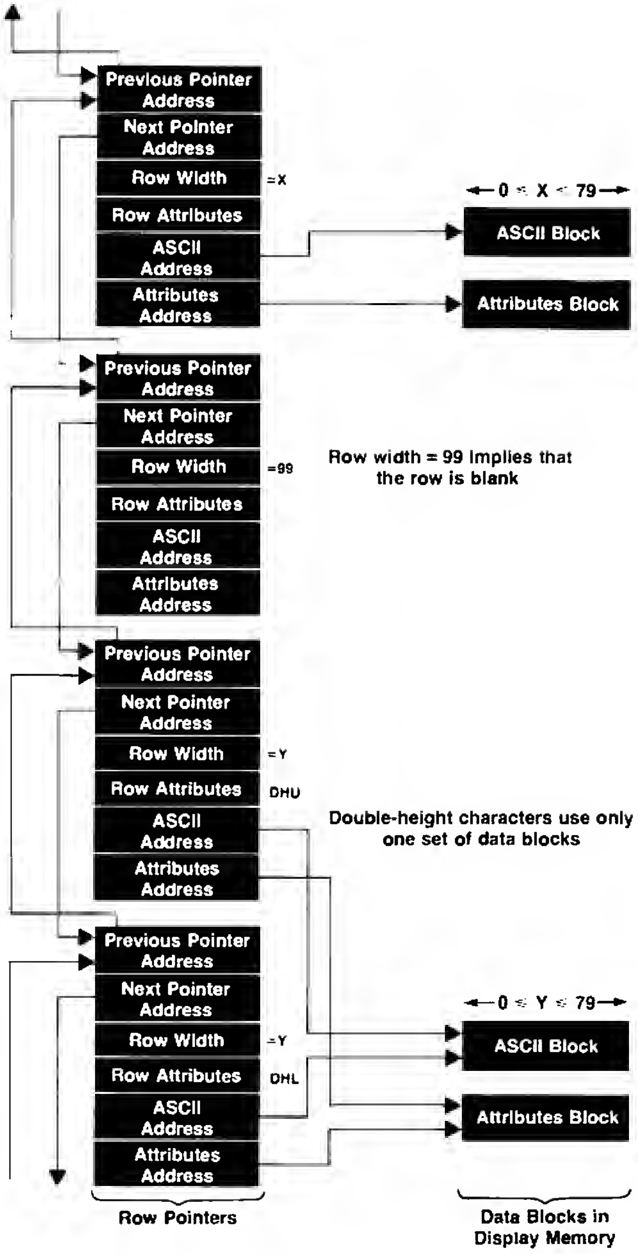

Fig. 2. *Characters to be displayed are specified by a linked list of pointers.*

The row attributes register contains information associated with each row of characters on the display (see Fig. 3). If bit 4 is not set, there will be no DMA cycle for attributes. If bit 3 is set, this row is part of the memory lock area on the screen. Bits 0, 1, and 2 determine the width and height of the characters in the current row. For double-height characters, two pointer blocks contain the same

*PLA = Programmable Logic Array
UART = Universal Asynchronous Receiver/Transmitter

Byte Number	Row Header Information
0	Previous Row Header Low Byte
1	Previous Row Header High Byte
2	Next Row Header Low Byte
3	Next Row Header High Byte
4	Row Width
5	Row Attributes **
6	Character Address Low Byte
7	Character Address High Byte
8	Attribute Address Low Byte
9	Attribute Address High Byte

**Row Attribute Bit Definition:
b7=b6=b5=not used by CRTC
b4=ATTP (attribute block present)
b3=NSS (not scrolling=lock row)
b2=DHU (double height upper part)
b1=DHL (double height lower part)
b0=DW (double width row)

Fig. 3. *Pointer block structure. The row attributes register contains information associated with each row on the display*

ASCII address (and attributes address, if needed). The first has bit 2 (DHU) set in the row attributes register to indicate the double-height upper part, and the second has bit 1 (DHL) set to indicate the double-height lower part. The ROM line generator uses this information to access the same information twice, in consecutive scan lines, so that the character is displayed two times taller than normal. For double width, bit 0 (DW) is set in the row attributes register, telling the master sequencer to read each character in the ASCII and attributes blocks twice. The character codes in the internal row buffer are then effectively doubled. The attributes logic on the chip can then double each dot in a scan line to make the character appear wider than normal. To do so, it uses the column number information, indicating the parity of the current character location on the screen, along with the DW bit.

The master sequencer in the CRTC is programmed to execute the DMA cycles, using the addresses found in each consecutive pointer block of the linked list of pointers. The DMA cycles are initiated at fixed time intervals by the CRTC, which asserts its BREQ (bus request) output. The microprocessor answers the handshake with its BACK (bus acknowledge) line, as soon as it completes execution of the current instruction. The use of DMA techniques for data transfer between RAM and the CRTC minimizes the amount of processor time needed to support the display. In the worst case, no more than 15% of the data and address bus bandwidth is used by the CRTC. The rest is available to the microprocessor.

CRTC Architecture

The CRTC is built around four internal buses, through which all the data and timing information is exchanged. Fig. 4 is the CRTC block diagram. The internal data bus and the address bus communicate with the external microprocessor data and address lines. The column number bus (CN bus) and the four-phase clock are exclusively internal. The sequence of operations and the flow of data in the chip are controlled by a master sequencer PLA using 18 inputs, 28 outputs, and 132 minterms. The master sequencer reads the contents of the register stack and the outputs of the column counter and uses them with its present state to determine its next sequence.

The register stack contains 16 registers, each 8 bits wide, an 8-bit incrementer, and a PLA to decode the instructions from the master sequencer. The register stack can input data from the internal data bus and output data to it. The outputs of the incrementer can be directed onto the internal address bus during a DMA operation. The instruction set of the stack includes simple operations between registers, such as increment, transfer, input, output, and NOP.

The column counter is a free-running toggle counter that outputs the value of the current character number across the screen, from 0 to 79. It counts through 109 to account for horizontal retrace.

The row buffer is a three-transistor dynamic RAM array of 2080 bits, organized as two banks of 80 13-bit words. Two DMA cycles are necessary to load one bank of the row buffer, one for ASCII data (8 bits) and one for attributes (5 bits). The words in the row buffer are addressed using the CN bus. As a result, read and write operations on this RAM are done synchronously with the display of the characters on the terminal screen. While the contents of one row buffer bank are read out to display on the screen, the second bank is written to, so that it is ready to be used for display after the current row of characters. At that time, the role of the two buffer banks is reversed.

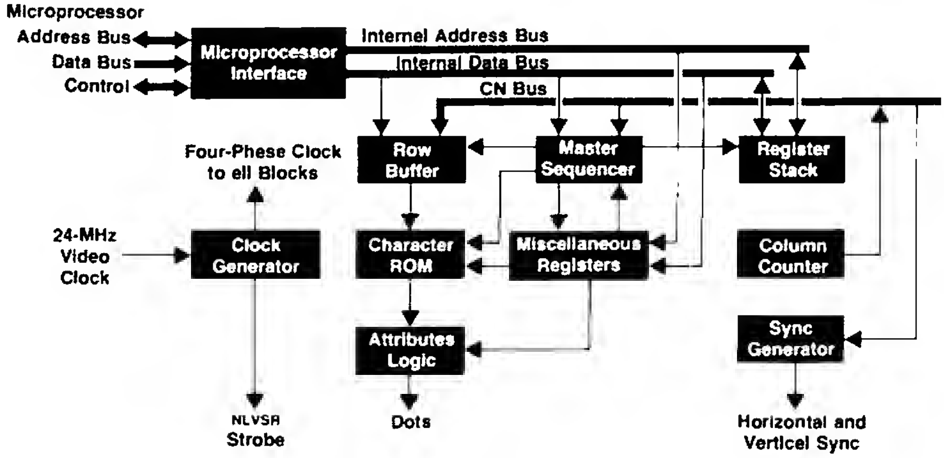

Fig. 4. *Block diagram of the VLSI CRT controller chip used in the HP 2392A Terminal.*

The character ROM is programmed with the dot patterns used to draw the characters on the screen, using a matrix 9 dots wide and 14 dots tall. The display process requires that the successive dot patterns forming one scan line be fetched and assembled into one continuous stream of dots, to be shifted at the video frequency (24 MHz). To display one scan line, the ROM addresses are formed by combining the scan line number from the register stack and the ASCII codes for the characters at each consecutive column number location. The ASCII codes come from the row buffer, addressed by the column counter output value. The information in one row buffer bank is used fourteen times, once for each scan line. Each time, a new scan line number is read from the register stack by the master sequencer. The master sequencer orders the stack to increment the scan line register during the horizontal retrace time.

The character ROM contains 256 character patterns. Each character is described as 14 groups of 8 bits for a total of 28,672 bits. The ROM cell and decoder designs were leveraged from an earlier HP 64K-bit ROM design.

The attributes logic block modifies the ROM output patterns according to the attribute bits for each character stored in the row buffer. Four attributes are processed by this block: inverse video, blinking, underline, and half bright. The cursor visible on the terminal screen is also added into the dot stream in this block, according to XY coordinates stored in one of the miscellaneous registers (part of the frame information written by the microprocessor during vertical retrace time).

Finally, the dot patterns corresponding to the characters to be displayed on the screen are output on the CRTC dot outputs as 9-bit words. A strobe signal (NLVSR) is provided by the clock generation block; this acts as the load enable for a fast gate array chip in which the dot shifting at the video frequency (24 MHz) takes place (see box, next page).

All of the CRTC blocks run synchronously on a four-phase nonoverlapped clock that has a period of 375 ns, equal to a character time, or 9 dots. This results in most of the chip running at 2.67 MHz. The register stack works at a 5.33-MHz rate, executing two operations within one basic four-phase cycle. The clock generation block inputs the 24-MHz video clock and derives the four-phase clocks and the NLVSR strobe signal from it. Although it would not be practical to make a whole chip work at such frequencies in the NMOS-C process, the performance of the process is sufficient for the small portion of logic necessary to achieve the correct phase relation between the 24-MHz main clock and the NLVSR strobe signal.

CRTC Layout

Fig. 5 shows the CRTC chip layout. The main blocks, such as ROM, RAM, PLAs, registers, and miscellaneous logic, are easily identified. One of the main advantages of VLSI is demonstrated here. The small ROM line generator block, which computes the effective scan line number used to address the ROM, would require a very large number of gates and a lot of board space if it were implemented in standard TTL logic. Designed as part of the chip using an array of pass transistors arranged in a barrel shifter, the whole function uses only about 1% of the total chip area. Operation of the ROM line generator is described in the box on page 15.

The total number of transistors on the CRTC chip is approximately 55,000. There are approximately 28,000 in ROM, 6000 in RAM, 12,000 in the PLAs, 2000 in the register stack, and 7000 in miscellaneous logic.

Design Tools

Being the first HP division to begin IC design outside the U.S.A., the Grenoble team had to face some unusual situations. Most of them were associated with limitations in the tools, which U.S. designers generally solved by going to a bigger, more powerful computer. Others required information exchange, and we could not fly to the U.S. every

(continued on page 14)

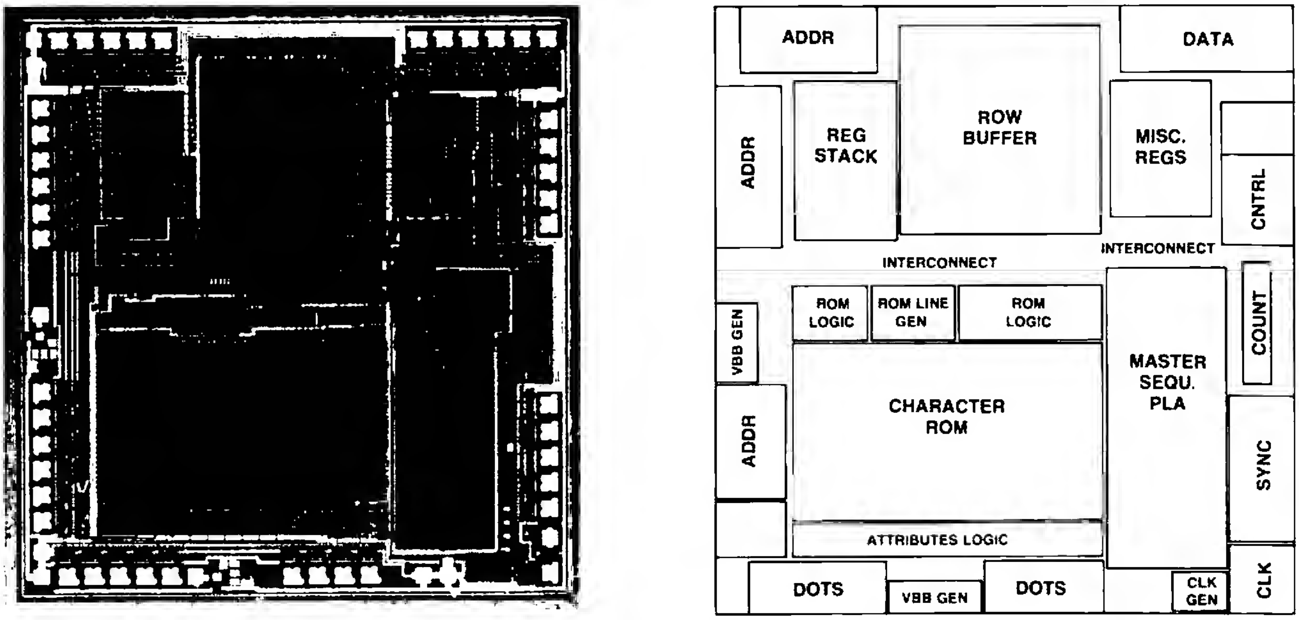

Fig. 5. *CRT controller chip and layout.*

A Fast Gate Array Companion for a CRT Controller

Because of the speed limitations of the NMOS technology used for the HP 2392A Terminal's CRT controller chip, the CRTC outputs, in parallel, the 9-dot pattern and the half bright attribute for the current scan line of the characters to be displayed. A separate gate array chip shifts these parallel dot patterns to a serial flow of bright and dark dots for the video amplifier input, and provides the gain control input of the video amplifier with the half bright attribute correctly synchronized to the first dot of each character, as shown in Fig. 1.

Standard HP character generators use half shifting capability to improve the character definition (one character line of dots may be shifted one half dot downstream), as shown in Fig. 2. In classical designs, this feature at certain times calls for twice the normal dot frequency, as shown in Fig. 3a. This effect can lead to marginal designs when standard Schottky TTL circuits are used. This need for high-frequency capability was avoided in the HP 2392A by using a slightly different shifting model where the unused 9th dot in half-shifted lines is dropped, which does not affect the visible result on the screen, as shown in Fig. 3b. The cost was a few additional gates. However, saving a few gates is not so important within a large gate array as it would be in a classical design using discrete ICs.

Gate Array Benefits

Cost reduction is one benefit of this gate array design, because the number of parts is reduced. An improvement in the reliability is also expected, both from the reduction in the number of parts and from the improvement of the design margins as explained above.

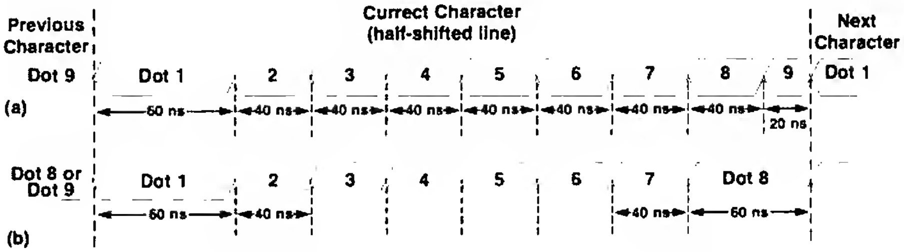

Fig. 3. *Classical half shifting model (a) calls for twice the normal dot frequency (note 20-ns dot). HP 2392A shifting model (b) does not place this demand on the CRT drive circuits.*

The high-frequency signals within the gate array are the main contributors to the EMI generated by the system. Their concentration within a single IC has a very positive impact on the EMI behavior of the terminal.

This video stage uses 40% of the available circuits within the chosen gate array (MCA500ALS from Motorola). The remaining circuits (Fig. 4) are used to reduce the logic board random logic (clock generation, UART interfacing, bell tone generator), thus providing additional board cost reduction, improvement of reliability, and improvement of the terminal's EMI performance.

This complete set of functions uses 100% of the available circuits. The complexity is equivalent to 18 to 20 SSI/MSI TTL circuits.

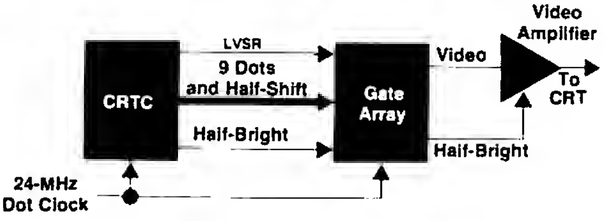

Fig. 1. *The gate array chip interfaces the CRT controller chip to the video amplifier.*

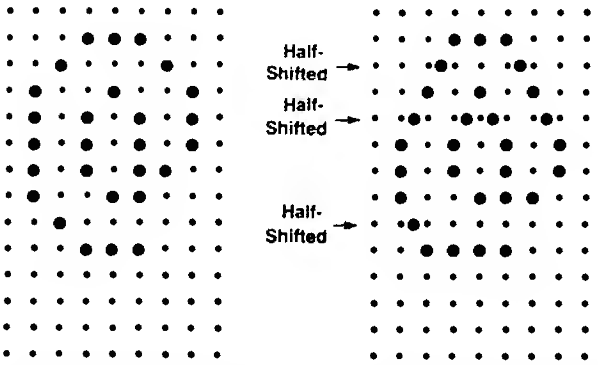

Fig. 2. *Half shifting is used to improve character definition. (left) @ character without half shift. (right) @ character with half shift.*

Design Stages

The MCA500ALS is a macrocell array, rather than a standard gate array. The internal part of the logic is divided into 24 blocks of elementary components (transistors, resistors, etc.).

During the first design step, a first level of interconnection from a library assigns each of these blocks to a defined and characterized logic function. For instance, one block may be converted to two D flip-flops, or to four 2:1 multiplexers.

This very rapid step to macro logic functions permits very precise preliminary investigations. Using 100% of the available

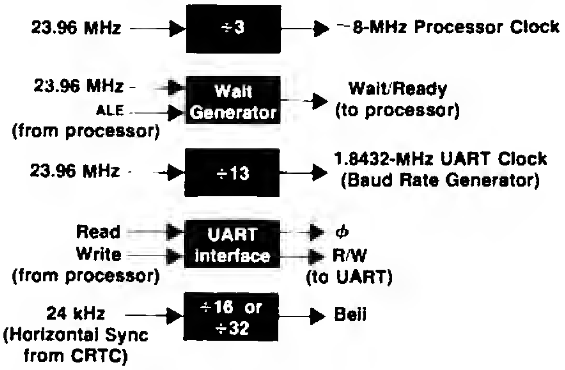

Fig. 4. *Besides the video serial register stage (Fig. 1), the gate array chip also provides the circuits shown here.*

hardware can be anticipated without any risk Accurate prediction of the dynamic performance is achievable from the characterization of the logic functions in the library.

In the second design step, the interconnections between these logic blocks are defined. At this step, the simulation capabilities of the development tools permit verification of the combinatorial logic of the circuit, estimation of the dynamic performance, taking into account the true load of each circuit, and evaluation of the metal length.

At the third design step, the physical positions of the 24 logic blocks and their physical interconnections are defined. Automatic placing and routing are available, but in our case it failed because 100% of the available circuits are used, so a very dense interconnection is necessary.

Placing was completely done by hand. Then autorouting drew all of the interconnections except one, at once. The last wire was placed by hand

When hand placing is used, automatic verification of some design rules is necessary. Detailed verification (static, functional, or dynamic) is also available, and the results correlated well with measurements performed 12 weeks later on the first samples of the circuit.

This design took eight engineer-days, after four days of training

Freddie Barbut
Development Engineer
Grenoble Personal Computer Division

(continued from page 12)

time we had a question to answer.

From the beginning, we had decided to rely on supported HP tools, both hardware and software. We used most of the tools available for schematic entry, simulation, artwork layout, extraction, and design rules checking.

When we started our design in early 1982, the only artwork layout systems available within HP were IGS on the HP 3000,[1] and Piglet on the HP 9845C.[2] We started our layout with both systems, allocating Piglet to cell and block design and the IGS station to layout and full-chip editing.

We used Draw[3] and Spice[3] on the HP 3000 to create schematics and perform electrical simulations of basic cells and blocks. At Grenoble, we do not have a link to a mainframe computer, so we relied totally on the HP tools. For example, Spice on the HP 3000 cannot deal with more than 30 to 40 transistors at a time. Therefore, only simple cells can be checked in one simulation pass. To overcome this limitation and be able to simulate the behavior of more complex signal paths, we wrote a program to create a piecewise-linear description of a Spice output. To check a signal path of, say, 80 transistors, our procedure was to split the schematic into three smaller schematics of 25 to 30 devices each, in such a way that each smaller block was driving a load equivalent to that in the original circuit. The first block was then simulated with Spice, a piecewise-linear approximation of the output was created and used as an input for simulation of the second block, and so on.

The design rules check of the full chip was also a problem. We learned very early in the development that the HP 3000 could not check a large design in one pass, so we performed design rules checks on overlapping windows, each covering a quarter of the chip.

Logic simulation and test vector generation were done with the Testaid program on the HP 1000.[3] Again, special programs were written to pass information between the data base on the HP 3000 and the HP 1000 system, simplifying the interface to Testaid and creating readable output listings on which the behavior of the circuit could be checked.

We were able to overcome the physical distances and the delays by resorting to yet another set of HP tools. We did not want to waste one to two weeks for mail delivery when sending the artwork archive tapes to Cupertino for manufacturing. Instead, we decided to use the data transfer capability of HP's worldwide Comsys system. We just had to store our archive data on a tape on our HP 3000 and give it to our Comsys operator in the next room. Two hours later, the data was available in the Comsys computer in Cupertino. The whole operation lasted less than half a day, and no tape ever traveled from Grenoble to Cupertino. In late 1982, when HP DeskManager (formerly HP Mail[4]) became widely available, we used it extensively to exchange not only information, but also programs and process data.

Evolution and Current Development

The NMOS-C process has evolved since we began designing the CRTC chip. In early 1984, a new, shrunk version of the process was made available to the designers. It brought the minimum channel length from 3 to 2.4 μm, using a linear 20% shrink on all dimensions. Designs to be fabricated in the new NMOS-CS process can be laid out using standard NMOS-C rules, and the 20% shrink is applied when manufacturing the masks. Of course, new process parameters are used for Spice simulations and circuit extraction, so that normal layouts are simulated as if they were shrunk, to check the behavior of the final circuit. The shrunk version of the process lets the same layout run approximately twice as fast as in the standard process, and the chip area is reduced by about 40%.

We plan to take advantage of this process evolution to design the successors of the CRTC chip. We are currently integrating on the CRTC all the functions that now reside in the fast gate array chip, such as the video shifter, the microprocessor clock and wait-state generation, and UART clock generation. The new NMOS-CS process allows us to design the 24-MHz shifters that could not be done safely in the standard version.

Acknowledgments

When the HP 2392A project began, the NMOS-C process had just been released, and we got the assurance of very good support from HP's Cupertino Integrated Circuits Operation. No one at HP in Grenoble had designed ICs at that time. I took the HP VLSI class while on assignment in the U.S.A. Then we hired John Connolly, an experienced HP IC designer. John became our class instructor for the first session of the VLSI class taught outside the U.S. in an HP division, and helped us gain the knowledge of IC design necessary to start our project. The design team included John Connolly, Richard Brabant, Khambao Panyasak, and

(continued on page 16)

How to Scroll Smoothly

In a classical video display terminal the rows of characters jump up or down when the roll up or roll down key is pressed This can cause eyestrain and fatigue for the operators The HP 2392A moves the rows up and down in a smooth linear motion The displacement is not really linear but actually consists of small jumps of two scan lines at a time on every frame time (16 milliseconds). as shown in Fig. 1

Fig 2 is a block diagram of the circuits involved in smooth scrolling

The master sequencer in the HP 2392A's CRT controller (CRTC) manages all video timing by updating the contents of the registers in the register stack (scan line number, row number. and frame number) The video display timing description is an absolute reference, as opposed to the description of the screen as viewed by the user, which is relative. When smooth scrolling is enabled, the relative and absolute values are not identical

The ROM line generator translates the absolute value of the scan line number into the relative matrix line number of the characters to be displayed The matrix line number called the ROM line, forms a four-bit part of the ROM address

The optional ANSI version of the HP 2392A implements the double-height feature Two bits in the row attributes part of the pointer blocks are used to designate double-height upper (DHU) and double-height lower (DHL) Because of this, the algorithm used to compute the relative line number has two forms

Normal height The ROM line address is the octal representation of the decimal sum of the absolute scan line number (SLN), and the smooth scrolling value (SS), modulo 14.

DH false: RL = (SLN + SS) mod 14

Double height. The ROM line address is the octal representation of the following (next page):

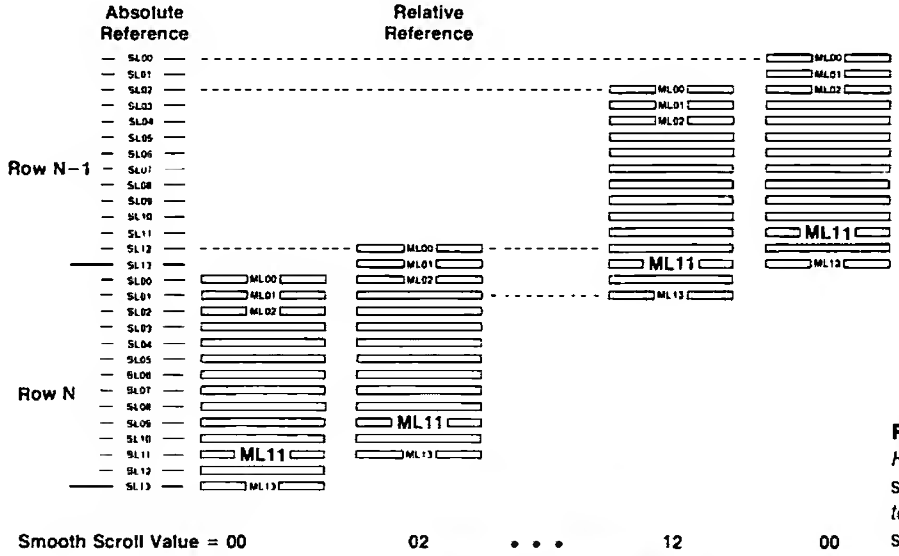

Fig. 1. *Smooth scrolling moves the HP 2392A display up or down two scan lines at a time, so the characters appear to move smoothly instead of jumping from row to row.*

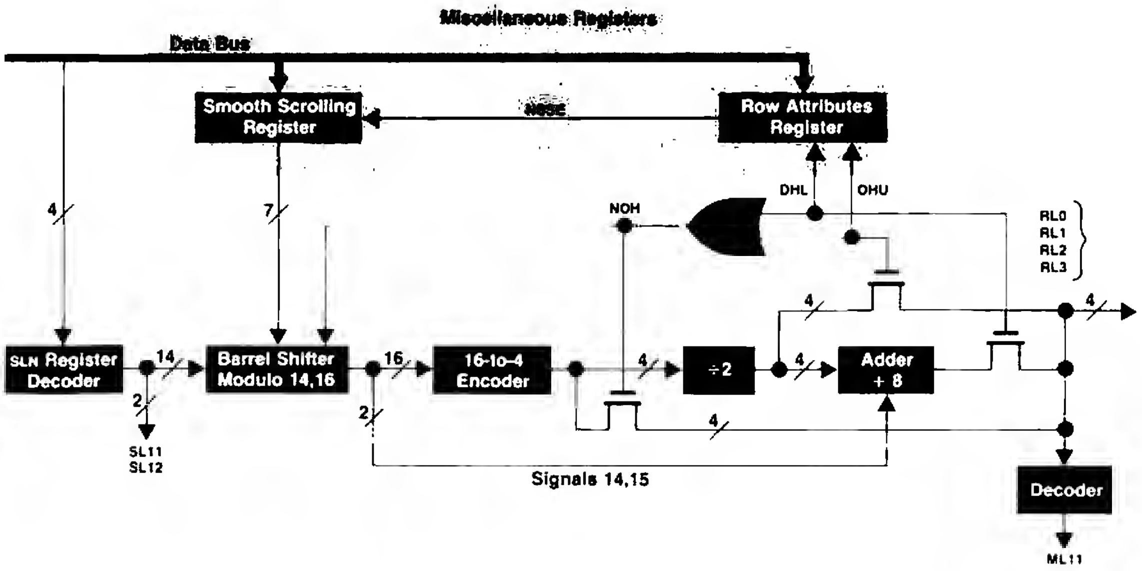

Fig. 2. *Smooth scrolling hardware diagram.*

```
DH true:  if DHU, RL = INTEGER[(SLN+SS) mod 16 / 2]

          if DHL and (SLN+SS)=14 or 15, then
                RL = INTEGER[(SLN+SS) mod 16 / 2]

          if DHL and [SLN+SS] ≠ 14 or 15, then
                RL = INTEGER[(SLN+SS) mod 16 / 2] + 8.
```

This results in the same ROM line being generated twice, so that each line of the character matrix appears twice, and the character is two times as large on the screen.

The ROM line generator also generates some signals used in the attributes logic. Matrix line 11 (ML11) is the relative line number in the character matrix that is turned on when the underline attribute bit is present. SL11 and SL12 are the addresses of the scan lines on which the cursor can be displayed.

Richard Brabant
Development Engineer
Grenoble Personal Computer Division

(continued from page 14)

myself for the design, and Guy Pascal for the mask layout. The design started in February 1982, and first prototypes came out in April 1983. Fully functional parts were available in November 1983, and ship release for the CRTC chip occurred in May 1984.

References

1. D.F. Bracken and W.J. McCalla, "An Interactive Graphics System for Structured Design of Integrated Circuits," *Hewlett-Packard Journal*, Vol. 32, no. 6, June 1981.
2. T.H. Baker, "IC Layout on a Desktop Computer," *ibid.*
3. L.K. Scheffer, R.I. Dowell, and R.M. Apte, "Design and Simulation of VLSI Circuits," *ibid.*
4. I.J. Fuller, "Electronic Mail for the Interactive Office," *Hewlett-Packard Journal*, Vol. 34, no. 2, February 1983.